

Backend

This is an example of how you may give instructions on setting up your project locally. To get a local copy up and running follow these simple example steps. Prerequisites

This is an example of how to list things you need to use the software and how to install them.

N.B: First install node js latest version. Then install npm install npm@latest -g.

Now go to the phase 3 directory. In this, you can see two folders server and client

First, open the server folder, and in the server directory, you can see the package.json file. In this file, all the dependencies are added. Open terminal in server folder and command “npm install”, That will do the package installation. If .env not found in server folder create .env file and add

```
{  
  
global_DB=mongodb+srv://mtrshuvo:v7qd1r5dO0T8bWz2@cluster0.zvn1z.mongodb.net/phase3?retryWrites=true&w=majority  
  
JWT_SECRET_KEY=SECRET  
  
NODE_ENV=development  
  
CLOUDINARY_CLOUD_NAME=dm9wer83l  
  
CLOUDINARY_API_KEY=566693299616286  
  
CLOUDINARY_API_SECRET=YrRoC3h1iNeV1bkpgc6DrenOA5w  
  
}
```

. After successfully the installation type another command “npm start” that will start the server. If successfully run, you should see the “successfully connected to DB”.

Frontend

First, open the client folder, and in the client directory, you can see the package.json file. In this file, all the dependencies are added. Open terminal in server folder and command “npm install”, That will do the package installation. If .env not found in server folder create .env file and add

```
{  
  
REACT_APP_API=http://localhost:3001/api  
  
}
```

.After successfully the installation type another command “npm start” that will start the client server. If successfully run, you should see the “successfully connected to DB”. For first time admin login userid 10001 and password 123456

Documentation for Phase 3 Project

Name: Touhedur Rahman

1. Sign in

Request

Type	Params	Values
Header	nsId	string
Body	password	string

nsID, the password must be in the request body.

Status	Response
200	{ "message": "Successfully Login", "data": { "_id": "62886948241d66a33f4781bd", "name": "nsladmin", "nslId": "10001", "role": "admin", "isFirstTimeLogin": false, "updatedAt": "2022-05-21T04:26:37.464Z" }, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1MDAwMCIsImVudCI6ImFkbm91dGVkaW5kdjEiLCJ1eWl1IjoiNzkzMdGxfQ.Wi5r52GKjbvHeazRNChwD_2bRpEFZ6dFu7T-Zq4FThc" }
400	{"error":"Incorrect username or password."}
500	{"error":"Something went wrong. Please try again later."}

Response

2. Create User

Create new user

Request

Method	URL
POST	api/users/createuser

Type	Params	Values
HEADERS	Authorization(token)	string
Body	Name	string
	nsId	string

The `token` must be in the Authorization headers section with Bearer. 'Bearer {token}'

Response

Status	Response
200	<pre>{ "message": "user created", "data": { "name": "Rony", "nsId": "10007", "password": "12345", "role": "user", "isFirstTimeLogin": true, "_id": "6289a7ef78c0197ac36135be", "createdAt": "2022-05-22T03:03:11.620Z", "updatedAt": "2022-05-22T03:03:11.620Z", "__v": 0 } }</pre>
401	{"message": "You are not authorize to create user account"}
500	{"message": "Account not create at this moment"}

3. user update

Update individual user.

Request

Method	URL
PUT	api/users/update

Type	Params	Values
HEADERS	Authoriazation(token)	string
Body	User info(password, name)	string

Response

Status	Response
200	<pre>{ "message": "Password Changed successfully", "data": { "_id": "62891684125aff920ab00c1e", "name": "roney", "ns1Id": "10006", "role": "user", "isFirstTimeLogin": true, "createdAt": "2022-05-21T16:42:44.410Z", "updatedAt": "2022-05-21T16:42:44.410Z", "__v": 0 } }</pre>
400	<pre>{"message": "wrong credential"}</pre>
500	<pre>{"message": "Password not changes"}</pre>

4. get all user

Get more information on a particular recipe

Request

Method	URL
GET	api/users/alluser

Type	Params	Values
HEADERS	Authoriazation(token)	string

Here Only role admin get the all user info.

Response

Status	Response
200	<pre>{ "0": { "_id": "62886c45c7cd3f63cb90efa1", "name": "Shuvo", "nsId": "10002", "role": "user", "isFirstTimeLogin": false, "createdAt": "2022-05-21T04:36:21.750Z", "updatedAt": "2022-05-21T04:41:24.479Z", "__v": 0 }, "1": { "_id": "62886d178d4dc45321244144", "name": "touhedur", "nsId": "10003", "role": "teamlead", "isFirstTimeLogin": false, "createdAt": "2022-05-21T04:39:51.366Z", "updatedAt": "2022-05-21T06:43:12.168Z", "__v": 0 }, }</pre>

	}
500	{"message": "Problem in fetch user"}

5. Project create

Create user by admin role

Request

Method	URL
Post	api/projects/create

Type	Params	Values
HEADERS	Authoriazation(token)	string
Body	title	string

Response

Status	Response
200	{ "message": "Project create successfully", "data": { "title": "Project two", "members": [], "_id": "6288756e0851eeca88f56a17", "createdAt": "2022-05-21T05:15:27.022Z", "updatedAt": "2022-05-21T05:15:27.022Z", "__v": 0 } }
400	{ "message": "Already title created" }

6. get single update

Documentation for Phase 3 Project

Name: Touhedur Rahman

Request

Method	URL
PUT	api/projects/:projectid

Type	Params	Values
HEADERS	Authoriazation(token)	string
Body	projectid	string

Response

Status	Response
200	<pre>{ "data": { "reports": [], "_id": "6288756e0851eeca88f56a17", "title": "Project two", "members": [{ "_id": "62886c45c7cd3f63cb90efa1", "name": "Shuvo" }], "createdAt": "2022-05-21T05:15:27.022Z", "updatedAt": "2022-05-21T08:44:20.454Z", "__v": 0, "teamLead": { "_id": "62886d178d4dc45321244144", "name": "touhedur" } } }</pre>
400	<pre>{ "message": "Project not found" }</pre>

7. update individual project

update individual project

Request

Documentation for Phase 3 Project

Name: Touhedur Rahman

Method	URL
PUT	/api/projects/:projectid

Type	Params	Values
HEADERS Body	Authoriazation(token) Title Members Start date End date Teamlead	string string [string] String String String(user_id)

Response

Status	Response
200	{ "message": "Updated successfully" }
400	{ "message": "Your are not authorize to update the information" }
500	{"message": "Password not changes"}

8. delete single project

Delete only by admin role

Request

Method	URL
POST	api/projects/:projectid

Type	Params	Values
HEADERS POST	Authorization(token)	string

Response

Status	Response
200	<pre>{ "message": "Updated successfully" }</pre>
400	<pre>{ "message": "Your are not authorize to delete the project" }</pre>

9. upload r&d report

Request

Method	URL
POST	/api/reports/upload

Type	Params	Values
HEADERS Body	Authoriazation(token) file	string string string

Response

Status	Response
200	<pre>{ "message": "file uploaded", "data": { "title": "exercise", "link": "https://res.cloudinary.com/dm9wer83l/image/upload/v1653143100/w6egehgxyizpzpmzz7u.pdf", "cloudinary_id": "w6egehgxyizpzpmzz7u", "_id": "6288f63dd802c3cbf78b854a", "__v": 0 } }</pre>
500	<pre>{"message": "Something went wrong"}</pre>

10.ChangePassword

Update individual user.

Request

Method	URL
POST	api/users/changepassword

Type	Params	Values
HEADERS	Authoriazation(token)	string
Body	User info(password, name)	string

Response

Status	Response
200	<pre>{ "message": "Password Changed successfully", "data": { "_id": "62891684125aff920ab00c1e", "name": "roney", "ns1Id": "10006", "password": "12345", "role": "user", "isFirstTimeLogin": true, "createdAt": "2022-05-21T16:42:44.410Z", "updatedAt": "2022-05-21T16:42:44.410Z", "__v": 0 } }</pre>
400	<pre>{"message": "wrong credential"}</pre>
500	<pre>{"message": "Password not changes"}</pre>

10.getAllReports

Update individual user.

Documentation for Phase 3 Project

Name: Touhedur Rahman

Request

Method	URL
POST	api/reports/

Type	Params	Values
HEADERS	Authoriazation(token)	string
Body	User info(password, name)	string

Response

Status	Response
200	<pre>[{ "_id": "6288f63dd802c3cbf78b854a", "title": "exercise", "link": "https://res.cloudinary.com/dm9wer83l/image/upload/v1653143100/w6egehgxyizpzpmzz7u.pdf", "cloudinary_id": "w6egehgxyizpzpmzz7u", "__v": 0 }, { "_id": "628a245b55e5fdfeefd36d01", "link": "https://res.cloudinary.com/dm9wer83l/image/upload/v1653220442/oaruf5fiwmvuzjcqsvei.pdf", "cloudinary_id": "oaruf5fiwmvuzjcqsvei", "__v": 0 }]</pre>
400	{"message": "no data found"}
500	{"message": "Error occured"}